

Jin-Soo Kim
jinsoo.kim@snu.ac.kr

Systems Software &
Architecture Lab.

Seoul National University

Jan. 6 – 17, 2020

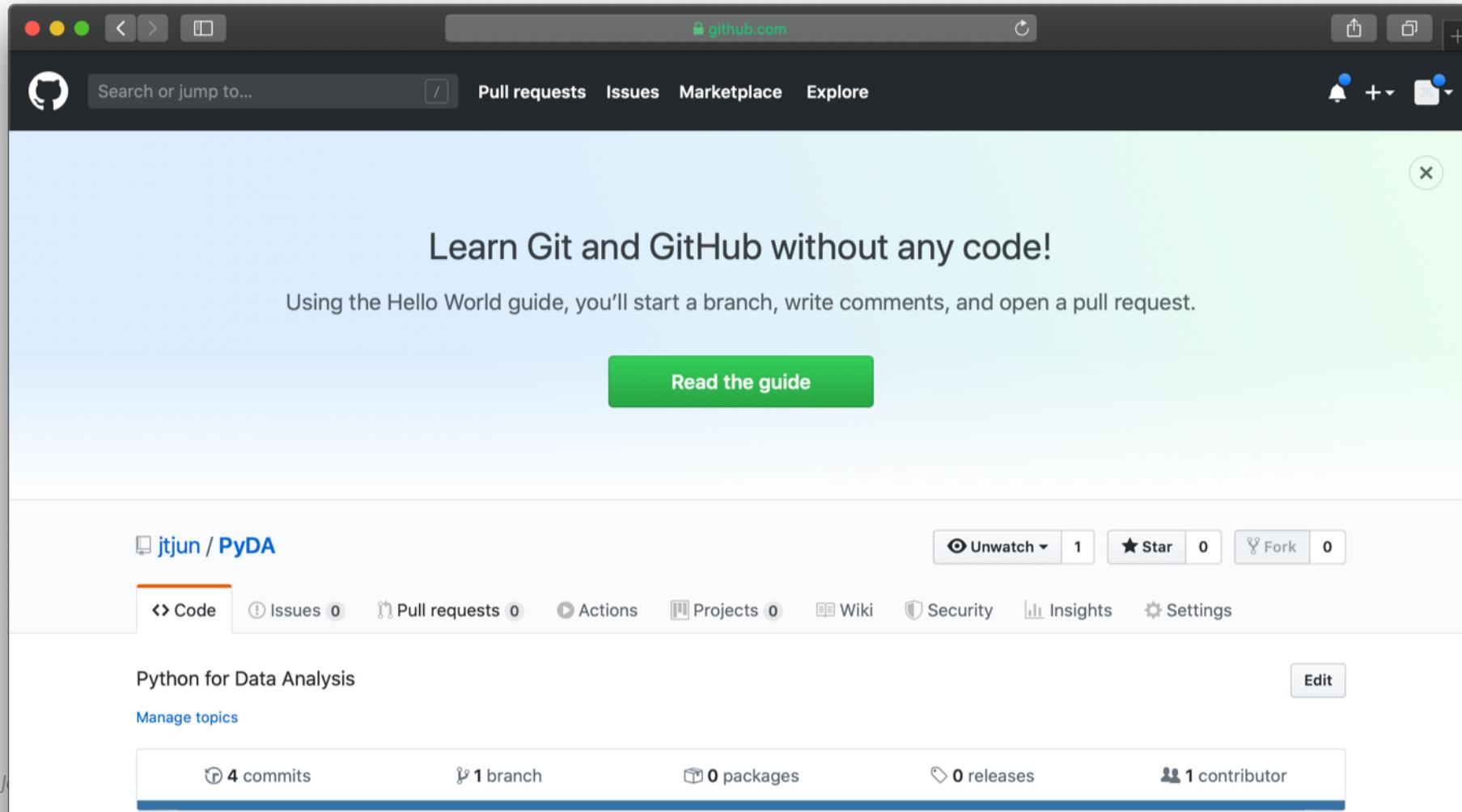


Python for Data Analytics

Python 0 I

다운로드 안내

- <https://github.com/jtjun/PyDA/archive/master.zip>



sys.snu.ac.kr : 실습 서버

■ Login & Register

The image displays two adjacent browser windows side-by-side, both showing the "SNU: Systems Software & Architecture Lab." website.

Left Window (Login Page):

- Header:** SNU: Systems Software & Architecture Lab.
- Title:** Login
- Fields:** Email (text input), Password (text input).
- Buttons:** Sign in (blue button), Join (gray button).
- Text at bottom:** We recommend the Google Chrome web browser.
© SNU Systems Software & Architecture Laboratory

Right Window (Join Page):

- Header:** sys.snu.ac.kr
- Title:** Join
- Fields:** Email (text input), Password (text input), Password Confirm (text input), Name (Real Name) (text input), NickName (text input), Student No. (text input), Mobile (text input - 010 - 0000 - 0000).
- Buttons:** Register (blue button), Cancel (gray button).
- Text at bottom:** We recommend the Google Chrome web browser.
© SNU Systems Software & Architecture Laboratory

sys.snu.ac.kr : 실습 서버

■ Main Page

The screenshot shows a web browser window for the URL `sys.snu.ac.kr`. The title bar reads "SNU: Systems Software & Architecture Lab.". The main content area displays information for the "[PyDA] Python for Data Analytics" class. On the left, a sidebar menu lists "CLASS LIST" (with "[PyDA] Python for Data Analytics" highlighted), "CLASS HOME" (with "Overview" highlighted), and "SYSTEM" sections. The "CLASS HOME" section includes links for "Projects", "Rank", "Latest Runs", "QnA", and "Course Homepage". The "SYSTEM" section includes "Test Queue Status", "Member Info", and "Logout". The main content area features a large heading "Python for Data Analytics" and a blue button "Goto Homepage ». Below this, a light blue box contains the text "[2020/01/08]" and "Word Count with List has been posted.", followed by the date "Date : 2019.01.07 13:30:00 ~ 2019.01.08 18:00:00". At the bottom of the page, there is a note about recommended browsers and copyright information.

SNU: Systems Software & Architecture Lab.

CLASS LIST
[PyDA] Python for Data Analytics

CLASS HOME
Overview

Projects
Rank
Latest Runs
QnA
Course Homepage

SYSTEM
Test Queue Status
Member Info
Logout

Python for Data Analytics

Goto Homepage »

[2020/01/08]

Word Count with List has been posted.

Date : 2019.01.07 13:30:00 ~ 2019.01.08 18:00:00

We recommend the Google Chrome web browser.

© SNU Systems Software & Architecture Laboratory

sys.snu.ac.kr : 실습 서버

■ Project Page

SNU: Systems Software & Architecture Lab.

CLASS LIST
[PyDA] Python for Data Analytics

CLASS HOME
Overview
Projects
Rank
Latest Runs
QnA
Course Homepage

SYSTEM
Test Queue Status
Member Info
Logout

We recommend the Google Chrome web browser.
© SNU Systems Software & Architecture Laboratory

SNU: Systems Software & Architecture Lab.

CLASS LIST
[PyDA] Python for Data Analytics

CLASS HOME
Overview
Projects
Rank
Latest Runs
QnA
Course Homepage

SYSTEM
Test Queue Status
Member Info
Logout

PyDA : Projects Submit

Word Count with List

dict 없이 list 만으로 단어의 빈도수 세기
Due date : 2019/01/08 18:00:00

[View details »](#) [Template Code »](#)

Source Code :
Report :

| # | Date | File | Compile | Execution | Final | Metric | Test | - |
|---------------|------|------|---------|-----------|-------|--------|------|---|
| No submission | | | | | | | | |

We recommend the Google Chrome web browser.
© SNU Systems Software & Architecture Laboratory

Word Count

Lab I-I. Word Count

- 주어진 파일 ()로부터 Database 를 구축하여 사용자가 '*EXITprogram*'을 입력할 때 까지 입력 받은 단어의 총 개수를 출력함.

```
$ python wordCnt.py
```

The ↵ 단어 하나만 입력

51 ↵ 빈도만 출력

The ↵

230

EXITprogram ↵

exit

Lab I-I. Word Count

■ Word Count DB 구축하기

- Parameter - path :
 - type : str
 - 읽고자 하는 파일의 경로
- Return - [words, cnts] :
 - type : list
 - [단어의 리스트, 빈도의 리스트] : 두 list 를 원소로 갖는 list
 - words 와 cnts 는 인덱스 별로 서로 대응함 (words[idx]의 빈도 = cnts[idx])

```
def word_cnt_db(path)
```

Lab I-I. Word Count

■ Word Count DB 구축하기 (cont'd)

- **Process :**
 - 주어진 파일에 있는 모든 unique 한 단어의 빈도를 계산
- **Caution**
 - file 열 때 오류 처리
 - 맨 끝에 있는 ''와 '\n' 같은 공백 처리
 - 맨 끝에 있는 문장 부호 처리 :, . ? : ; ‘ “
 - 두 개가 동시에 오는 경우도 고려 ('This is end of line.\n')

```
def word_cnt_db(path)
```

Lab I-I. Word Count

- 입출력 형식을 맞춰 interact 하기

- Parameter - path :
 - type : str
 - 읽고자 하는 파일의 경로
- Return - None:
 - type : None
 - 단, 마지막에 exit(0) 로 프로그램 종료
- 6 page 의 입출력 예시 참고

```
def word_cnt_list(path)
```

Lab I-I. Word Count

■ 입출력 형식을 맞춰 interact 하기 (cont'd)

- **Process:**

- 사용자가 ‘*EXITprogram*’을 입력할 때 까지 입력 받은 단어의 총 개수를 출력

- **Caution:**

- 사용자의 입력을 받을 때, 아무런 문구를 띄우지 않음

- 즉, `input()` 함수의 parameter로 아무것도 넣지 않음

- 출력 시 오직 빈도와 ‘*\n*’ 하나만 출력

- 즉, `print(5)` 또는 `print('5\n', end = '')` 식으로 출력 (words에 없는 경우 → 0)

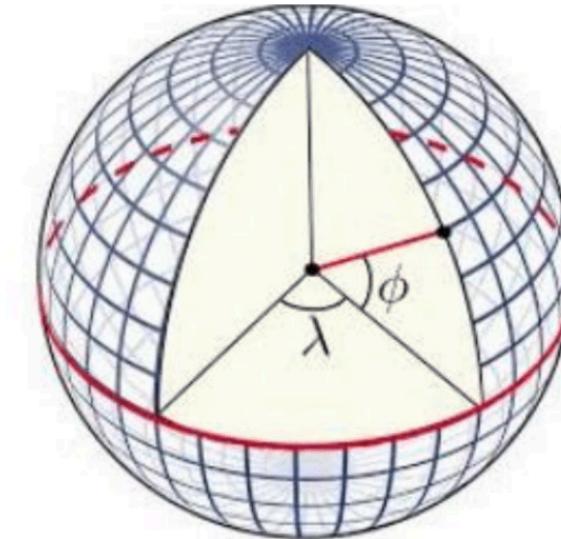
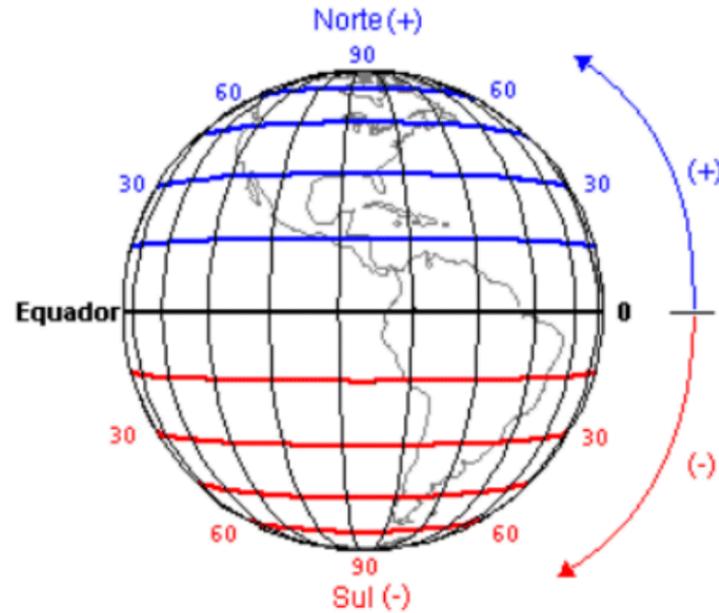
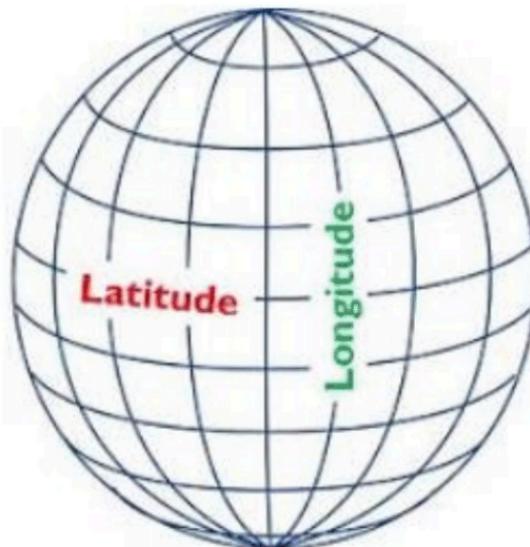
```
def word_cnt_list(path)
```

Cities

Lab I-2. Cities

■ 지구 상에서 두 지점간 거리?

- 지구는 둥그니까~
- 직선 거리가 직선 거리가 아니다.
- 구 위의 임의의 두 점의 거리는?



Lab I-2. Cities

- 지구 상에서 두 지점간 거리?

- 코드로 표현하는 방법?
- math 라이브러리를 사용!
- import math → math.pi, math.sin, math.cos, math.acos

$(lat1, lon1), (lat2, lon2)$

$$\begin{aligned} \cos \theta &= \sin(lat1)\sin(lat2) + \cos(lat1)\cos(lat2)\cos(lon2 - lon1) \\ \theta &= \text{acos}(\cos(\theta)) \end{aligned}$$

$$distance = r\theta, \quad 2\pi \text{ radian} = 360^\circ$$

Radian 각도로 표현된 두 지점 사이 거리

Lab I-2. Cities

- 사용자에게 두 위치를 받아서 거리를 출력하기

```
def radian(d)

def dist_between_loc(loc1, loc2)

def loc_dist()
```

Lab I-2. Cities

■ 각도를 radian 형태로 변환하기

- Parameter - d :
 - type : float
 - 변환하고자 하는 각도의 360° 형태
- Return - radian:
 - type : float
 - radian 형태로 변환된 각도
 - Hint. $360^\circ = 2\pi$

```
def radian(d)
```

Lab I-2. Cities

■ 두 지점 간의 거리 구하기

- Parameter – loc1, loc2 :
 - type : list [float, float]
 - latitude, longitude 가 순서대로 담긴 두 개의 list
- Return - dist:
 - type : float
 - 앞의 공식을 이용해 계산한 거리
 - 단, lon2 – lon1 은 degree form에서 계산 (radian 변환 전에 계산 후 변환)

```
def dist_between_loc(loc1, loc2)
```

Lab I-2. Cities

■ 사용자의 입력 처리하기

- Parameter - None :
 - 단, 사용자에게 input 을 받음 (Word Count 와 같이 input() 함수의 인자는 없음)
 - 띄어쓰기로 구분된 'lat lon'을 두 번 입력 받음
- Return - None:
 - 값을 return 하는 것이 아닌, 사용자에게 보여줌
 - 다른 것 없이 거리만 출력
- 18 page의 입출력 예시 참고

```
def loc_dist()
```

Lab I-2. Cities

- 여기에서 집까지 거리 구하기

```
$ python distEarth.py  
37.455813 126.954699 ↵ 첫 번째 위도 경도  
37.266653 126.999386 ↵ 두 번째 위도 경도  
21401.21251958583 ← 거리  
: 컴퓨터 연구소 - 수원역
```



Lab I-2. Cities

- 파일에서 도시 정보 읽기

- 도시 정보 모아둔 파일 `cities.txt`
- 도시의 이름, 도시의 나라\t도시의 위도\t도시의 경도
- 파일을 읽어서 DB 구축하기
- Word Count 에서 했듯이 두 개의 list 이용 [`names`, `locs`]

```
def get_cities()
```

Lab I-2. Cities

■ 구축한 DB에서 필요한 정보 찾기

- `get_cities()` 함수를 통해 .txt에 있는 정보를 프로그램으로 가져옴
- 내가 원하는 정보를 찾아내는 함수 만들기
- 도시의 이름을 넣어서
 - 해당 도시가 있는 경우 위치 좌표 반환 (위도, 경도)
 - 없는 경우 `None` 반환

```
def get_city_by_name(cities, name)
```

get_cities() 함수에서 반환 받은 list [names, locs]
찾으려는 도시의 이름 str

Lab I-2. Cities

■ 사용자의 요구에 따라 거리 알려주기

- 도시 정보 모아둔 파일 `cities.txt` 에 있는 도시 이름 두 개를 넣었을 경우 (순서에 상관 없이) 거리를 출력
- 이전에 구현한 함수들을 활용
- 단, `get_cities()` 함수는 전체 코드에서 단 한 번만 호출
- 22 page 의 입출력 예시 참고 : 종료 조건, 예외 처리

```
def city_dist()
```

Lab I-2. Cities

■ 서울에서 평양 거리

```
$ python cities.py  
Seoul, Pyeongyang ↵ ← ';'로 두 도시 구분  
239024.51825296393 (white space 주의)  
a, Seoul ↵  
There are no 'a'  
Paris, so far ↵  
There are no 'so far'  
EXITprogram ↵  
exit
```

왜 오차가 발생할까요?

195 km

서울특별시에서 평양까지의 거리

